AMENDMENTS TO THE SPECIFICATION:

Please replace paragraph [0053] on page 12 with the following amended paragraph:

[0053] In one embodiment, running estimates are generated of processor utilization for each encoding level based upon measured processor utilization. Referring to Figure 7, in one embodiment CPU usage controller 655 acquires input control parameters 705 at the start of a time interval 710. CPU usage controller 655 acquires CPU usage measurements 715 for a current encoding level as part of its process of deciding whether to adjust the encoding level 725. The instantaneous CPU usage measurements of the software encoder, $\{E_i\}$, are used to generate dynamic estimates, $\{\overline{E}_{n,i}; 0 \leq n < N_{max}\}$, for each encoding level of the encoder n and for each time interval (or iteration) i. In one embodiment, initial CPU usage estimates, $\{\overline{E}_{n,0}; 0 \leq n < N_{max}\}$ are calculated for each encoding level and then updated 720 for each new i 722 recursively according to the formula:

$$\overline{E}_{n,i} = \begin{cases} \alpha \cdot \overline{E}_{n,i-1} + (1-\alpha) \cdot E_i & \text{if } n = n_{curr} \\ \overline{E}_{n,i-1} & \text{otherwise} \end{cases},$$

where n_{curr} is the current encoding level of the encoder and α is an aging factor that controls the rate at which the estimate can change. In one embodiment, the estimates are further forced to be monotonic _ (i.e., $\overline{E}_{n,i} \geq \overline{E}_{n-1,i}$).

Please replace paragraph [0055] on pages 12-13 with the following amended paragraph:

[0055] Figure 8 is a flowchart illustrating one embodiment of a method of incrementing and decrementing encoder levels. CPU usage controller 655 responds to deviations in the CPU usage measurements E_i and I_i by dynamically decrementing or incrementing the encoding level n of the encoder at the completion of each time interval i. The encoding level n is decremented 815 to n = n - 1 in response to determining that the following three conditions 820 are met:

$$I_i < I_{min} ,$$

$$E_i > E_{min} ,$$

$$n > 0 ,$$

The encoding level is incremented 835 in response to determining 830 that the following two conditions are met:

$$I_i > I_{min} + \Delta_i ,$$

$$n < N_{max}$$

where Δ_i is the expected increase in CPU usage that will result if the encoding level is increased and is given by:

$$\Delta_i = \overline{E}_{n+1,i} - \overline{E}_{n,i} .$$

Because the estimates for CPU usage increase monotonically with encoding level, Δ_i is assured to be non-negative, and therefore the conditions for decrementing and incrementing the encoding level are mutually exclusive. Note that in one embodiment that the first startup adjustment 805 is to select a target startup level 810 corresponding to the lowest ($n = 0$) encoding level.

Please replace paragraph [0067] on page 16 with the following amended paragraph:

[0067] As one example, there are a variety of transcoding applications in which a user may desire as high a throughput as possible but also want to run other software programs on the computer while the transcoding process is being performed. Transcoding is used, for example, to ~~convet~~ <u>convert</u> a recorded show or song to be viewed on another type of device (e.g., converting from the format of a laptop to that of a handheld pocket PC type device). Transcoding may be used to reduce the storage required for a show or song. Transcoding also may be used to store a show or song onto another type of medium, such as a CD. In offline transcoding a user may schedule conversion (transcoding) of a recorded television show or song from one format to another format or from a higher bit-rate to a lower bit-rate. The conversion

process may take a significant amount of time, depending upon the data size of the file being converted, such that it is scheduled in the background. The user typically desires that the conversion is completed as fast as possible, i.e., at the highest possible throughput. In accordance with one embodiment of the present invention, a transcoding program has at least two different performance levels corresponding to two different tradeoffs between processor utilization and throughput. In analogy to previously discussed examples, the processor usage and idle thread utilization are monitoring and the performance level of the transcoding program is selected to be within control constraints such that other programs have sufficient idle thread utilization to start and execute properly.